

REMARKS

Claims 1, 3, 5 - 12, 30 - 33, 35, and 37 - 49 are pending. Claims 1, 3, 5 - 7, 9, 11 - 12, 30 - 33, and 35 have been amended. Claims 37 - 49 have been added. Claims 2, 4, 13 - 29, 34, and 36 have been cancelled. No new matter has been introduced.

Reexamination and reconsideration of this application are respectfully requested.

In the April 4, 2004 Office Action, the Examiner objected to the title of the invention. The applicants have amended the title of the invention to be clearly indicative of the invention to which the claims are directed.

In the April 7, 2004 Office Action, the Examiner identified that claims 1- 36 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,112,320 to Dien ("the Dien reference") in view of U.S. Patent No. 6,434,612 to Hughes et al. ("the Hughes reference"). This rejection is respectfully traversed in so far as they are applicable to the pending claims.

Independent claim 30, as amended, recites:

A method of managing a plurality of applications, said method comprising:

executing a first application, at least a portion of the first application being executed by a first computer and at least a second portion of the first application being executed at a user terminal, the first computer and the user terminal being coupled via a network;

executing a second application at the user terminal;

collecting, by a keep-alive function, first application timeout information related to said first application;

formulating a keep-alive input based on the collected timeout information; and

transmitting said keep-alive input from said user terminal to said first computer.

The Dien reference does not disclose, teach, or suggest the method of independent claim 30, as amended. The Examiner states that the Dien reference

discloses a computer-executable keep-alive function hosted by the user terminal, where the keep-alive function is configured to cause the user terminal to transmit a message to the first computer within a period based on the timeout period. (*Office Action, page 3*). The applicants respectfully disagree with the Examiner. The Dien reference is directed to a watchdog timer for a computer (only one computer) with a CPU and a peripheral controller. (*Dien, Abstract, col. 2, lines 34 - 47*). In other words, there is no disclosure of a user terminal and a first computer coupled via a network, as recited in independent claim 30, as amended, because the Dien reference is referring to a single computer and not a network environment.

The Dien reference also does not disclose executing a first application, at least a portion of the first application being executed by a first computer and at least a second portion of the first application being executed at a user terminal.

The Dien reference does not disclose applications being executed, instead the Dien reference discloses that a watchdog function is enabled in a peripheral controller to send a check command to CPU and to interrupt the CPU periodically through an interrupt output . Further the Dien reference discloses that when interrupted, the watchdog interrupt service routine returns watchdog check data to the peripheral controller 2. This information is either execution counters or other program counters of the CPU as updated by other programs. The watchdog program in the peripheral controller then decides whether the CPU is operating normally or not. If the CPU does not respond at all, the watchdog program will generate a reset on the reset output and reboot the system. The Dien reference also discloses that the CPU can check the status of the peripheral controller and reset the peripheral controller if it does not like

the response it receives. (*Dien, col. 2, line 48 - col. 3, line 19*). In addition, there is no disclosure that at least a portion of the first application is executed on a first computer and a second portion of the first application is executed on a user terminal, as is recited by independent claim 30, as amended.

Further, the Dien reference does not disclose **collecting, by a keep-alive function, first application timeout information related to said first application; and formulating a keep-alive input based on the collected timeout information**. The Dien reference discloses that the watchdog program collects watchdog check data from the CPU, which could include the execution counter or the program counters. These counters allow the watchdog program to determine if the CPU (and thus the system) is running correctly. (*Dien, col. 2, lines 48 - 65*). This is not the same as **timeout information related to the first application** because Dien does not disclose the retrieving of time out information. Instead, it is focused on CPU execution counters and program counters. In addition, **a keep-alive input cannot be formulated** because the Dien reference does not collect timeout information. Assuming, *arguendo*, that the Dien reference did collect timeout information, the Dien reference does not disclose formulating a keep-alive input because the Dien reference discloses only the generation of a reset for either the CPU or the peripheral controller. A reset is not the same and actually the opposite of a keepalive input. Accordingly, applicants respectfully submit that independent claim 30, as amended, distinguishes over the Dien reference.

The Hughes reference does not make up for the deficiencies of the Dien reference. The Examiner states that the Hughes reference discloses a computer-executable time-out function hosted by the first computer, the timeout function to cause

the first computer to terminate the application if a keep-alive input is not received by the first computer within a timeout period. (*Office Action, page 3*). The Examiner also states that the application timeout clock is reset when the first computer receives a keep-alive input, and the timeout information includes information related to a current state of the timeout clock. (*Office Action, page 4*).

Specifically, the Hughes reference discloses that a controller requests that a switch create, delete, or change connections. The switch may accept or reject the requests based on resource availability. The switch responds to the controller with a message indicating the outcome and the controller is required to get an explicit reply for all requests and take action if the response is not received within a time-out period. (*Hughes, col. 5, line 60 - col. 6, line 6*). The Hughes reference also discloses that when a user configures a switch, the configured switch activates the control port and all communication channels to connect slaves (in a switch) to a master (in a controller) via the control port. Upon connection, the slaves send switch information traps to the master with the session identification. The user configures the controller to active the control port driver with the port, controller number, and a number of slaves for the switch. Upon configuration, the controller activates the port and all communication channels to all the possible slaves. Upon connection to the slaves, the controller actives keep-alive signals to each slave. The controller detects slaves through traps and keep-alive responses. The keep-alive message is sent when no messages have been received from the slave in the configured time-out period. Detection of a slave keep-alive time out results in the master marking the slave as being unavailable. The slave may respond to other keep-alive message or may send traps to the master to

show that it is recovered. (*Hughes, col. 14, line 50 - col. 15, line 13*).

The Hughes reference, alone or in combination, with the Dien reference does not disclose a method of managing a plurality of applications, said method including **executing a first application, at least a portion of the first application being executed by a first computer and at least a second portion of the first application being executed at a user terminal**, the first computer and the user terminal being coupled via a network; **executing a second application** at the user terminal; and **collecting, by a keep-alive function, first application timeout information related to said first application**. The Hughes reference is directed to creating, deleting, or changing connections between a switch and a controller. The Hughes does not mention or disclose executing of a first application or a second application. The Hughes reference also does not disclose that **a portion of the first application is executed by a first computer and a second is executed by a user terminal**. In other words, there is no disclosure that an application co-resides on the controller and the switch.

In addition, assuming, *arguendo*, that a first application is disclosed by the Hughes reference, the Hughes reference does not disclose **collecting, by a keep-alive function, first application timeout information related to the first application**. The Examiner never identifies that any portion of the Hughes reference discloses this limitation. Instead, the Hughes reference discloses that controller transmits keep-alive messages at the end of a time-out period and the slaves respond to these keep-alive messages. This is not the same as the collection of timeout information from a first application because the Hughes

reference is describing only that a keep-alive message is sent at the end of a time-out period and independent claim 30, as amended, recites that a timeout function collects time-out period information (such as the actual time-out period to be utilized).

Further, because the Hughes reference does not disclose the collection of time-out information, there can be no disclosure of **the formulation of a keep-alive input based on the collected timeout information**. In fact, the Hughes reference discloses that the time-out period is configured by a user, not **formulated from the collection of time-out information of a first application**. Accordingly, applicants respectfully submit the claim 30, as amended, distinguishes over the Hughes reference, alone or in combination with the Dien reference.

Independent claim 1, as amended, recites limitations similar to independent claim 30, as amended. Accordingly, applicants respectfully submit that independent claim 1, as amended, distinguishes over the Dien and the Hughes references, alone or in combination, for similar reasons as those discussed above in regard

Claims 3, 5, 6 - 12, and 43 - 49 depend, directly or indirectly, on claim 1, as amended. Claims 31 - 33, 35, and 37 - 42 depend, directly or indirectly, on claim 30, as amended. Accordingly, applicants respectfully submit that claims 3, 5, 6 - 12, 31 - 33, 35, and 37 - 49 distinguish over the Dien and the Hughes references, alone or in combination, for the same reasons as those discussed above in regard to independent claims 1 and 30.

Claims 35 and 37 further distinguishes over the Dien and Hughes reference, alone or in combination. Claims 35 and 37 recite:

35. The method of claim 30, wherein said application timeout information includes information related to a **required content for said keep-alive input**, and said keep alive input is formulated based on said application timeout information.

37. The method of claim 30, wherein said application timeout information includes information related to a **required format for said keep-alive input**, and said keep-alive input is formulated based on said application timeout information.

Neither the Dien reference nor the Hughes reference disclose that the application timeout information includes information related to a required content or format for a keep-alive input. The Dien reference discloses only that the watchdog check data may include program counters or other program counters, which the watchdog timer utilizes to determine whether the system is running normally or not. (*Dien, col. 2, line 48 - 65*). This is not related to a **required content or format for a keep-alive input**. Accordingly, applicants respectfully submit that claims 35 and 37 further distinguish over the Dien and the Hughes references, alone or in combination.

Claim 5 recites limitations similar to claims 35 and 37. Accordingly, applicants respectfully submit that claim 5 further distinguishes over the Dien and the Hughes reference, alone or in combination, for reasons similar to those discussed above in regard to claims 35 and 37.

Claim 38 distinguishes over the Dien and the Hughes references, alone or in combination. Claim 38 recites:

The method of claim 30, further including **transmitting a keep-alive information request to the first computer in order to collect the first application timeout information**.

There is no disclosure in either the Dien or Hughes references that a **keep-alive information request** is sent to collect the first application timeout information. Instead, the Dien reference discloses only transmitting a check request to a CPU or to a controller and the Hughes reference discloses only transmitting a keep-alive request.

These keep-alive requests (or check requests) are only awaiting a reply and are not the same as a keep-alive information request. Accordingly, applicants respectfully submit that claim 38 further distinguishes over the Dien and Hughes references, alone or in combination.

Dependent claim 46 recites limitations similar to claim 38. According, applicants respectfully submit that claim 46 distinguishes over the Dien and Hughes references, alone or in combination, for similar reasons as those discussed above in regard to claim 38.

///

///

///

///

///

///

///

///

///

///

///

///

///

///

///

Applicants believe that the foregoing amendments place the application in condition for allowance, and a favorable action is respectfully requested. If for any reason the Examiner finds the application other than in condition for allowance, the Examiner is requested to call either of the undersigned attorneys at the Los Angeles telephone number (213) 488-7100 to discuss the steps necessary for placing the application in condition for allowance should the Examiner believe that such a telephone conference would advance prosecution of the application.

Respectfully submitted,

PILLSBURY WINTHROP LLP

Date: October 7, 2004

By: Mark R. Kendrick
Mark R. Kendrick
Registration No. 48,468
Attorney for Applicants

Date: October 7, 2004

By: Roger R. Wise
Roger R. Wise
Registration No. 31,204
Attorney For Applicants

725 South Figueroa Street, Suite 2800
Los Angeles, CA 90017-5406
Telephone: (213) 488-7100
Facsimile: (213) 629-1033